Advanced Logic

http://www-sop.inria.fr/members/Martin.Avanzini/teaching/2022/AL/

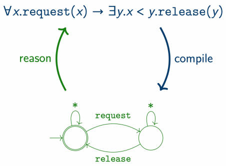
Martin Avanzini (martin.avanzini@inria.fr)
Etienne Lozes (etienne.lozes@univ-cotedazur.fr)

MASTER INFORMATIQUE
UNIVERSITÉ CÔTE D'AZUR

2nd Semester M1, 2022

Last Lectures

We saw how it is possible to synthesize a finite state automaton from a formal specification written in WMSO logic.



BUT sometimes one does not even know how to specify the ”thing”

Examples: security policy based on log analysis, program synthesis, invariant synthesis, etc

TODAY we will see how to synthesize a finite state automaton by learning it from examples and counter-examples.

Today's Lecture : L^* Algorithm

INFORMATION AND COMPUTATION **75**, 87–106 (1987)

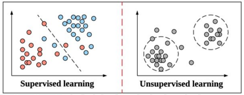
Learning Regular Sets from Queries and Counterexamples*

DANA ANGLUIN

Department of Computer Science, Yale University, P.O. Box 2158, Yale Station, New Haven, Connecticut 06520

The problem of identifying an unknown regular set from examples of its members and nonmembers is addressed. It is assumed that the regular set is presented by a *minimally adequate Teacher*, which can answer membership queries about the set and can also test a conjecture and indicate whether it is equal to the unknown set and provide a counterexample if not. (A counterexample is a string in the symmetric difference of the correct set and the conjectured set.) A learning algorithm L^* is described that correctly learns any regular set from any minimally adequate Teacher in time polynomial in the number of states of the minimum dfa for the set and the maximum length of any counterexample provided by the Teacher. It is shown that in a stochastic setting the ability of the Teacher to test conjectures may be replaced by a random sampling oracle, *EX*( ). A polynomial-time learning algorithm is shown for a particular problem of context-free language identification.

Dana Angluin



the Learner and the Teacher

Learner

Ignores L

Learns A s.t. $L = L(A)$

query $\Longrightarrow$

$\Longleftarrow$ answer

Teacher

Knows L

Answers all queries without mistakes

- ⋆ membership query: $w \in L$? Answer: yes/no.
- ⋆ conjecture query: $L = L(A)$? Answer: yes/no, because w is a counter-example.

Observation Table

An observation table is a tuple (S, E, T) such that:

- ⋆ $S = \{u_1, \ldots, u_n\} \subseteq \Sigma^*$ is a finite, prefix-closed set of words ("starters")
- ⋆ $E = \{v_1, \ldots, v_m\} \subseteq \Sigma^*$ is a finite, suffix-closed set of words ("enders")
- ⋆ $T : (S \cup S.\Sigma) \times E \to \{0, 1\}$ is the table

Intuition
$T(u, v) = 1$ if and only if $uv \in L$

Representation

- ⋆ one column per ender,
- ⋆ one "upper" row per starter,
- ⋆ one "lower" row per starter+letter pair

Example: $\Sigma = \{a, b\}$, $S = \{\epsilon, a, aa\}$
$E = \{\epsilon, ab, b\}$, $L = a^* b$

	ϵ	ab	b
ϵ	0	1	1
a	0	1	1
aa	0	1	1
b	1	0	0
ab	1	0	0
aaa	0	1	1
aab	1	0	0

Row Promotion

The lower row of ux is covered by the upper row u' if $\text{row}(ux) = \text{row}(u')$

If a lower row is not covered, the learner promotes it to an upper row.
This may introduce new lower rows, and the learner performs membership queries to fill them.

	ϵ	ab	b
ϵ	0	1	1
a	0	1	1
aa	0	1	1
b	1	0	0
ab	1	0	0
aaa	0	1	1
aab	1	0	0

Rows b, ab, and aab are not covered

	ϵ	ab	b
ϵ	0	1	1
a	0	1	1
b	1	0	0
aa	0	1	1
ab	1	0	0
aaa	0	1	1
aab	1	0	0
ba	0	0	0
bb	0	0	0

After the promotion of row b, the rows ab and aab are covered, and the new lower rows ba, and bb are filled by membership queries ($L = a^* b$)

Closed Table

If all lower rows are covered, the table is called closed.

	ϵ	ab	b
ϵ	0	1	1
a	0	1	1
b	1	0	0
aa	0	1	1
ab	1	0	0
aaa	0	1	1
aab	1	0	0
ba	0	0	0
bb	0	0	0

The table is not closed because ba and bb are not covered

	ϵ	ab	b
ϵ	0	1	1
a	0	1	1
b	1	0	0
ba	0	0	0
aa	0	1	1
ab	1	0	0
aaa	0	1	1
aab	1	0	0
bb	0	0	0

After the promotion of ba, the table is closed

Column Insertion

Two upper rows u, u' are similar if $\mathrm{row}(u) = \mathrm{row}(u')$

Two upper rows u, u' are distinguished by letter x if $\mathrm{row}(ux) \neq \mathrm{row}(u'x)$

When u, u' are similar but distinguishable, the learner chooses $x \in \Sigma$ and $v \in E$ such that $T(ux, v) \neq T(u'x, v)$ and inserts a column xv in the table (filling it by asking membership queries)

Exemple: $L = a(a+b)aaa$

	ϵ	aaa	aa	a
ϵ	0	0	0	0
a	0	0	0	0
aa	0	1	0	0
b	0	**0**	0	0
ab	0	**1**	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

ϵ and a are similar
but distinguished by b

	ϵ	aaa	$baaa$	aa	a
ϵ	0	0	**0**	0	0
a	0	0	**1**	0	0
aa	0	1	0	0	0
b	0	0	0	0	0
ab	0	1	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

column $baaa$ has been inserted
ϵ and a are not similar any more

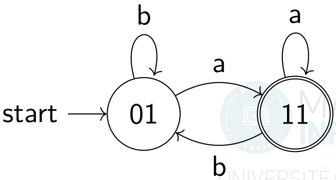
Automaton Defined by a Consistent Closed Table

A table is consistent if all similar rows are indistiguishable.

A consistent, closed table (S, E, T) defines a DFA (Q, δ, q_I, F):

	ϵ	a
ϵ	0	1
a	1	1
b	0	1
aa	1	1
ab	0	1
ba	1	1
bb	0	1

- ⋆ $Q = \{\text{row}(u) \mid u \in S\}$
  the states are the bitvectors appearing on the upper rows
- ⋆ $\delta(\text{row}(u), x) = \text{row}(ux)$
  well-defined only when the table is consistent and closed
  WHY?
- ⋆ $q_I = \text{row}(\epsilon)$
- ⋆ $F = \{\text{row(u)} \mid \text{T}(\text{u}, \epsilon) = 1\}$



L^* Algorithm

1. Start with $S = E = \{\epsilon\}$
2. repeat row promotion/column insertion until T is consistent and closed
3. submit A computed from T to the teacher
4. if the teacher gives $w \in L \Delta L(A)$, add w and its prefixes to S, and go to 2.
5. otherwise, return A

Demo: see `https://fissored.github.io/TER-M1-S2/`

Properties of the Algorithm

Correctness: if A is returned, then $L = L(A)$

proof: the teacher does not make any error!

Termination: if L is a regular language, then an automaton A is eventually returned

proof: next slides

Minimality: any A submitted to the teacher is a minimal DFA

proof (sketched): if two rows r_1, r_2 differ in column u, then u is accepted by A either starting from state r_1, or state r_2, but not both, so the two states r_1 and r_2 cannot be merged
reminder: the mininal DFA is obtained from any DFA through quotienting by bisimilarity

Towards Proving Termination : Notion of Residual

Given a language L and a word u, we write $u^{-1}L$ to denote the language of words that are obtained from the words of L starting with u by erasing their prefix u

$$u^{-1}L \stackrel{\text{def}}{=} \{v \mid uv \in L\}$$

Example: take $L = (a + bb)^*$, then $(ab)^{-1}L = b(a + bb)^*$

$u^{-1}L$ is called a residual of L.
We will write $\mathrm{Res}(L)$ to denote the set of residuals of L, i.e. $\mathrm{Res}(L) = \{u^{-1}L \mid u \in \Sigma^*\}$.

Example: $\mathrm{Res}(L) = \{L, b(a + bb)^*, \varnothing\}$. Indeed:

$$\epsilon^{-1}L = L \qquad \begin{array}{l} a^{-1}L = L \\ b^{-1}L = b(a+bb)^* \end{array} \qquad \begin{array}{l} (aa)^{-1}L = L \\ (ab)^{-1}L = b(a+bb)^* \\ (ba)^{-1}L = \varnothing \\ (bb)^{-1}L = L \end{array} \qquad \cdots$$

Towards Proving Termination : Recognizability and Residuals

Theorem a language L is DFA-recognizable if and only if $\mathrm{Res}(L)$ is finite.

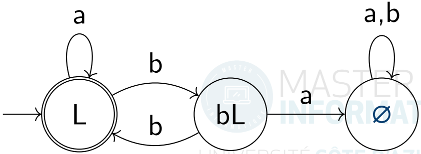
proof of left to right implication

Assume $L = L(A)$ for some DFA $A = (Q, \delta, q_I, F)$. Let $q(u) = \delta^*(q_I, u)$ be the state reached after reading u. Then $u^{-1}L$ is the language accepted by $A[q(u)]$ (starting from $q(u)$ instead of q_I). So $\mathrm{Res(L)} = \{L(A[q]) \mid q \text{ reachable from } q_I\}$, therefore $\sharp\mathrm{Res}(L) \leq \sharp Q < \infty$.

proof of right to left implication

Assume $\mathrm{Res}(L) = \{L_1, \ldots, L_n\}$. Take $Q = \mathrm{Res}(L)$, $\delta(L_i, x) = x^{-1}L_i$ (wich is a residual), $q_I = L$, and $F = \{L_i \mid \epsilon \in L_i\}$. Then $A = (Q, \delta, q_I, F)$ recognizes L.

Example: $L = (a + bb)^*$, $\mathrm{Res}(L) = \{L, bL, \varnothing\}$, $A =$



Towards Proving Termination : Residuals and Tables

Let a table (S, E, T) be fixed, and let $\sim_T$ be the equivalence of residuals defined by

$$L_1 \sim_T L_2 \qquad \text{if} \qquad L_1 \cap E = L_2 \cap E$$

Lemma 1: $\text{row}(u) \ = \ \text{row}(v)$ if and only if $u^{-1}L \sim_T v^{-1}L$

proof: by definition, $\text{row}(s)$ contains 1 in column e if and only if $e \in s^{-1}L$.

Lemma 2: Let A_L denote the minimal DFA accepting L, i.e. the automaton of residuals. Then

$$A_T = A_L / \sim_T$$

Proof of Termination of L^*

Let A_1, A_2, $\ldots$ denote the sequence of conjectures made by L^*. When a conjecture is rejected, at least one new row is added to the table, therefore

$$\sharp A_1 < \sharp A_2 < \ldots$$

Since all of these are bounded by $\sharp A_L$, the sequence of conjectures is finite.
To end the proof, observe that the "table completion" procedure also terminates for any table. Indeed, if $(S, E, T) \to (S', E', T')$ corresponds to a row promotion or column insertion, then the number of disimilar rows strictly increases. But $\mathrm{row}(u) \neq \mathrm{row}(v)$ implies $u^{-1}L \neq v^{-1}L$, therefore the number of disimilar rows is bounded by the $\sharp A_L$, which ends the proof.

Some final remarks

Complexity L^* learns A_L in time $O(mn^2)$, where n is $\sharp A_L$ and m is the maximal length of a counter-example given by the teacher (details in the article)

In practice the teacher may not know A_L either, and answering the conjecture queries may be based on some heuristics (like sampling)

Programming Assignment 2 use L^* to define a function that learns an automaton A from a WMSO specification